Trigger Files Text Analysis and Processing Time Anomaly Detection

Josué Irizarry Rodríguez
Master of Engineering in Computer Engineering
Jeffrey L. Duffany, Ph.D.
Electrical and Computer Engineering and Computer Science Department
Polytechnic University of Puerto Rico

Abstract *— Today the data processing is a vital part of the system supporting organization processes and transactions. A lot of data is managed, transmitted and stored daily. The format of the data is structured, unstructured or semi structured. Information Technology researchers are all in agreement that unstructured data is at least 80 percent of all enterprise data. It is unrealistic to expect that data will be perfect. The unstructured data hides important, when it is analyzed will aid businesses to have better decisions. The proposal of this project is an application using machine learning techniques like, information retrieval, data mining, and text mining with the intention to find possible anomalous values or processing time delays in the archived trigger files corpus.*

Key Terms *— Data Mining, Machine Learning, Text Mining, Trigger File.*

INTRODUCTION

Today is commonly the intercommunication between systems to perform dedicated tasks. The system interfaces permit to connect heterogeneous platforms to perform tasks. The labeling system refers to a text processing system loose coupling interface design. The text processing system starts the operation when a text including static and variable data called trigger file is dropped in a specified folder. The trigger file is the input to perform the system operation to print the label. The text processing systems are an asynchronous interface, receiving inputs from multiple stations. The throughput of the text processing system depends of a FIFO (first in, first out) queue, database iteration, network usage, server CPU and Memory usage, but not limited to any other factors that can cause delay in the processing time. The development of machine learning models will help to detect unseen processing time delays or anomaly in trigger files values. Without machine learning will be very difficult or impossible to detect outliers. This document presents design information, using software engineering techniques learned in the Mastery in Software Engineering. The agile software development is the software engineering used in this project.

PROBLEM STATEMENT

New system configurations or code changes in the software generating the trigger files can put at risk the normal operation or label output integrity. System interface and software code changes require carefully managed; otherwise it could result in a wrong output. For example, the new version of software or configuration changes could result in unexpected system behaviors, incorrects or anomalous system outputs. Nowadays, using the existing tools and techniques will very difficult to detect any an anomaly in trigger file or any processing time delay. It is important to mention the text processing system could process daily 8 thousand trigger files on average. Traditional data analysis is human dependable, requiring to spend a lot of hours looking into data or reports trying to find patterns or outliers.

PROJECT GOALS

The main goal of this project is to provide a short term, cost effective solution for anomaly detection in the text process system data processing or processing time delays. The data source for these machine learning models is a semi-structured trigger file format. The trigger file includes fixed fields and variable information in an attribute key and value format. These machine learning models were developed using Python programming language using existing packages for data mining,

information retrieval and text mining. For anomalous detection is using distance methods [3] like Cosine similarity, Levenshtein distance and TF-IDF. Someone argues Python, an open source language, is displacing R as language for data analysis. Python software applications are designed to run on client computers and servers, able to run on multiple platforms: UNIX, Linux, Windows and Mac. There is a lot of demand on jobs for persons that know Python. There are dedicated job search web pages like http://www.pythonjobs.com/ or at popular job search web pages such Monster.com or Dice.com.

The objective of this project includes the design and development of a text analysis system using machine learning algorithms capable for the anomaly detection. The models proposed for this project are documents anomaly detection in set of document, terms anomaly detection, text classification and processing time control chart (SPC) [13]. Related to the system processing, delays can be observed representing it as outlier values using time series chart [4]. Trigger file raw data can hide important information. The label system is processing in average of 8 thousands trigger files daily. Representation a lot the efforts for each label verification/inspection. See Figure 1 for a sampling of trigger files processed by day.

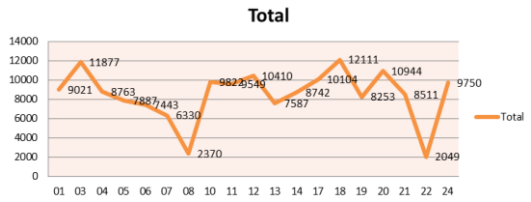

Figure 1
Quantity of Trigger Files Processed by Day

HIGH LEVEL EXPLICATION OF TEXT PROCESSING SYSTEM

The text processing system receives the required information from a text file called trigger file. On demand, every time a trigger data is copied in an input folder where the interface starts the process, print the label and perform archiving of the trigger file in a designed location with a different name and adding processing information. In the figure 2 presents the workflow diagram of the text processing interface system used for the label generation process.

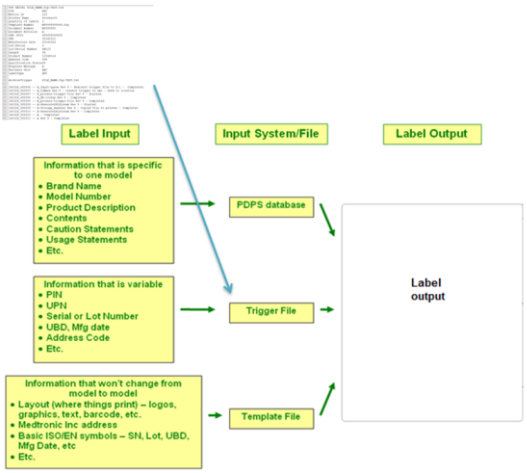

Figure 2
Text Processing System Inputs, Processing and Output

BRIEF REVIEW OF THE LITERATURE

Machine Learning

Machine learning, considered a subfield of computer science and statistics to build computer systems capable learn from data. Machine learning ties to artificial intelligence and optimization. The intention of Machine learning is getting computers to act without being explicitly programmed [10].

Data Mining

Data Mining is defined a technology mix traditional data analysis methods with advanced algorithm to process large quantities of data. By definition Data Mining is an automatic process used to discover useful information in large data set [11]. The intention is to find useful patterns for the organization that using another technique can remain unknown or hidden. Data Mining is subfield of the computer science. It involves method at the intersection of artificial intelligence, machine learning, statistics and database system. Data Mining is a very important component of knowledge discovery in database (KDD). KDD is the overall process of convert of taking data into useful information. The data mining process

consists of a series of transformation steps, from data preprocessing to post processing of data mining results.

The data mining tasks are commonly divided into two major categories:

1. Predictive tasks. The objective is to predict the value of an attribute based on the values of other attributes. Target or dependent are called the attribute to be predicted, while explanatory or independent variable is called for the attributes used for marking the prediction.
2. Descriptive tasks: The objective of this task is to derive patterns (correlations, trend, clusters and anomalies) summarize underlying relationships in data.

The data mining/text mining algorithms used in this project are clustering analysis and anomaly detection. The intention of the cluster [5] analysis will be seeking to find groups of closely observations part of same cluster are more like to each other than observations part of other clusters. Anomaly detection is the task in charge of the identification of the observations whose are significantly different from the rest of the data. A good anomaly detector must have a high detection rate and a low false alarm rate. The challenges of the anomaly detection are: number of the outliers found in the data, method is unsupervised [1] mean the validation cans enough challenging (such as for clustering [5]). K-means clustering techniques permit to data partition in each observation corresponding to the cluster with nearest mean. [8]. In case of this project the K-Means algorithm will be used to create clusters of text mined data.

Text Mining

Text mining [2] [10] consists of the analysis data existing in natural language text. Text analytics is application of text mining techniques to solve business problems. Text mining can help organizations to get potentially valuable information from text-based content like email, word documents and social network data. Mining semi-structured data is possible thanks to natural processing language algorithms (NLP) [6] [9], statistical modeling and machine learning are challenging. The ambiguities of the data caused by inconsistent syntax and semantics. Text analytics is considered as an emerging technology. [7]

Algorithm Supervised Learning

Supervised learning is the machine learning task of inferring a function from labeled training data. [1] The training data consist of a set of training examples. In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value. A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances. See Figure 3.

Unsupervised Learning

Unsupervised learning is any algorithm used to find hidden structure or patterns in unlabeled data. Since the examples given to the learner are unlabeled, it is very possible there is no error or reward signal to evaluate a potential solution. See figure 3.

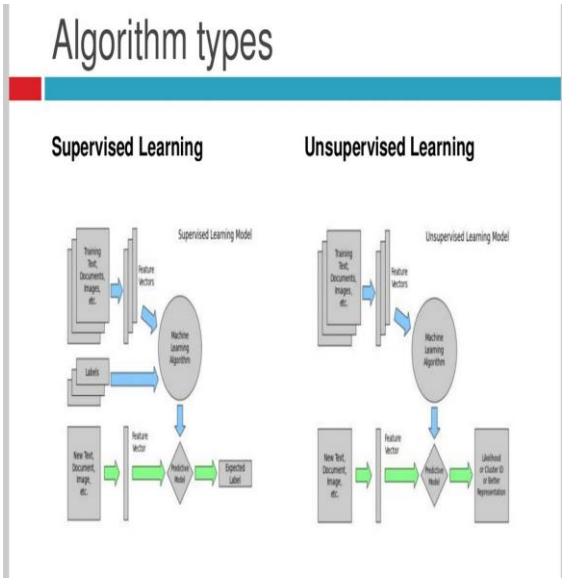

Figure 3
Algorithm Types

MACHINE LEARNING TECHNIQUES USED

The following table below describes the machine learning techniques used in this project.

Table 1
Machine Learning Techniques

Technique	Short description	Learning Type (Supervised, Unsupervised)
Information Retrieval	Information retrieval (IR) is retrieval terms, including term frequency, document frequency and inverse document frequency from semi-structured trigger files.	Unsupervised
Clustering analysis - K Nearest Neighbors	Cluster analysis groups of data objects based only on information found in the data that describes the objects and their relations. With the goal of finding or visualized related (similar) object from anomalous objects.	Unsupervised
Multinomial Naive Bayes	A Naïve Bayes Classifier assumes conditional independence between the random variables that constitute the features – not always true; still useful It can use historical data to create a probabilistic model, useful for estimating the likelihood of a particular classification given an incomplete set of known attributes.	Supervised
Linear SVC	Construct a linear SVM (Support Vector Machine) classifier. It is considered an extremely fast machine learning algorithm for classification in a large data set.	Supervised
TF-IDF	The utilization of the TF-IDF (term frequency-inverse document frequency) permit to find terms (words) less common in a corpus.	Supervised/ Unsupervised
NLTK	Basic functions of NLTK (Natural Language Processing with Python) used Levenshtein distance to support terms anomaly detection.	Unsupervised
Time series	Visualization of the data processing time	Unsupervised
SPC (Statistical Process Control)	Using basic function of Statistical process control (SPC) of Control Charts to determinate if the processing time of the trigger file is inline or outline of the Upper Control Limit. Outlier value is the one outside of the UCL.	Unsupervised
Cosine similarity	Used to determinate text document clustering and possible document outliers.	Unsupervised
Jaccard index similar	Used to know the Jaccard index similarly of between two structure serial	Supervised

	number strings.	
OneClassSVM	SVM models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis.	Supervised
Levenshtein distance	Levenshtein distance or edit distance is used to measure the edited characters between pairwise strings.	Unsupervised

SYSTEM ARCHITECTURE

This section provides information about the system architecture: System Process Workflow diagram, support software, models, database tables, GUI design and supporting software used.

Supporting Software

- Python 2.7.6
  - Enthought Canopy IDE 1.4.1
  - PyCharm IDE 3.4
  - Sklearn package 0.15
  - PySide for Front End 1.2.2
  - NLTK 3.0
- MySQL RDBMS version 5.6.20
- Visualization:
  - Matplotlib, for charts creation of Python code 1.3.1
  - Third party Business Intelligence software Pentaho 5.1

System Process Workflow Diagram

The system consists of model to break the consolidated archived trigger files into individual files, the machine learning models are one for Terms Anomaly Detection, Documents Outlier Detection, a model for Terms Anomaly Detection, a model for Text Classification using Naïve Bayes and LinearSVC and a model for the anomaly elapsed processing time using a Control chart (SPC). The diagram presents the module to write summarized information to the database and has available interaction with third party Business Intelligence applications. The diagram below describes the system operation workflow.

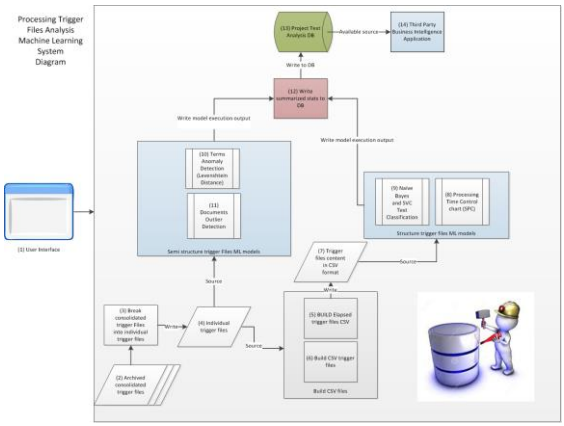

Figure 4
System Process Workflow Diagram

Refer to Table 2 for the modules that support the text analysis project.

Table 2
System Process Modules

Mapping	Machine learning module	Action performed
1	User Interface	User interface to execute with the Machine Learning models.
2	Archived consolidated trigger files	The source of the trigger files information consolidated in one file by day.
3	Break consolidated Trigger Files to individual trigger files	The trigger files content can be in individual files or into consolidated files for archived trigger files. This piece of code is in charge to break the consolidated file into a single trigger file using on the opening and ending tags that delimited each trigger file into the consolidated file. Individual trigger files will created when this module is executed.
4	Individual Trigger Files	Individual trigger files will be the sources of some models.
5	Build Elapsed Trigger File CSV	Script to perform Information retrieval of the trigger file processing time information and write out this information about the execution step with date time in a CSV format.

6	Build CSV Trigger Files	Script to perform Information retrieval of the trigger file features information and write out this information in a CSV format. Excluded the process information.
7	Trigger File's content in CSV format	In memory merge of the CSV files: features information with processing time.
8	Processing Time Control chart (SPC)	This model is in charge to determinate the Mean, Upper Control Limit, Lower Control Limit using SPC to determinate is the time processing a trigger file by system is inline or it is considered outlier. The model will perform an execution by Printer name and also for the label Template number. DB table: model_processing_time
9	Machine Learning model using Naïve Bayes and SVC for Text Classification	Model to predict the label based on the other trigger file attribute. The score of the two models are reported into the table model_text_classification.
10	Terms Anomaly Detection	The terms anomaly detection will measure Weighted Levenshtein distance in pairwise of a list of products and term attribute name. In the Levenshtein distance edit in an alphabetic the portion of the containing the string will have a greater weight compare to changes in digit part of the string. Using this design permits to don't detect as an outlier normal changes in the serial number sequences. For the purpose of this model the weighted levenshtein distance result greater or equal to one will be considered as an outlier term. The analyzed attribute name, attribute value, weighted levenshtein distance, record count, field format, length of the field, the standard deviation for the length of the term belonging same attribute name from semi-structured trigger files are stored into the MySQL. End users can perform term search query of the term using MySQL table: model_term_clustering. See Figure 11. This output information will include the terms considered anomalous, in which files were located and score of the model. DB table: model_term_clustering
11	Documents outlier detection	The module will read every trigger file. Cosine distance is used to determinate the degree of similarity of the documents. See Figure 5. This model requires training of the data. The execution statistics of the model for each document comparison will be saved in a DB. The information table will include information such start date and end time, name of the document compare, name of the master document (first read file) where the comparison is performed against, cosine distance results, flag is comparison is considered anomalous file is belonging to lower dynamic line, if the prediction of cosine distance is less to the established threshold, and another flag based on the result of the prediction. Also the table contains important information about the differences of the text and document and master document text content. DB table: model_document_clustering
12	Write summarized stats to DB	Write the analyzed information to a MySQL DB.
13	Project Text Analysis DB	MySQL Database design.
14	Third Party Business Intelligence Application	Visualization of the analyzed information from Business Intelligence application like Pentaho.

```
arreglo = cosine_similarity(tfidf_matrix_train[0:1], tfidf_matrix_train)
```

Figure 5
Compute Document Cosine Distance

Database Design

The total of five tables has been designed to support this project. MySQL is the database selected for this project. Refer to the Table 2 Description with mapping for system process workflow where is mapped the MySQL tables are

used and also Table 3 DB tables – models. A DB user account is used to perform the data manipulation and work for the design of the tables. A read only DB user account is available for third party applications to retrieve data. See Figure 6 for the database design diagram.

Table 3
DB Tables – Models

Table	Model supporting
Model_processing_time	Processing time Control Chart (SPC)
Model_term_clustering	Terms Anomaly Detection
Model_document_cluster	Documents Outlier Detection
Model_text_classification	Text classification
Models_stats	General model execution results

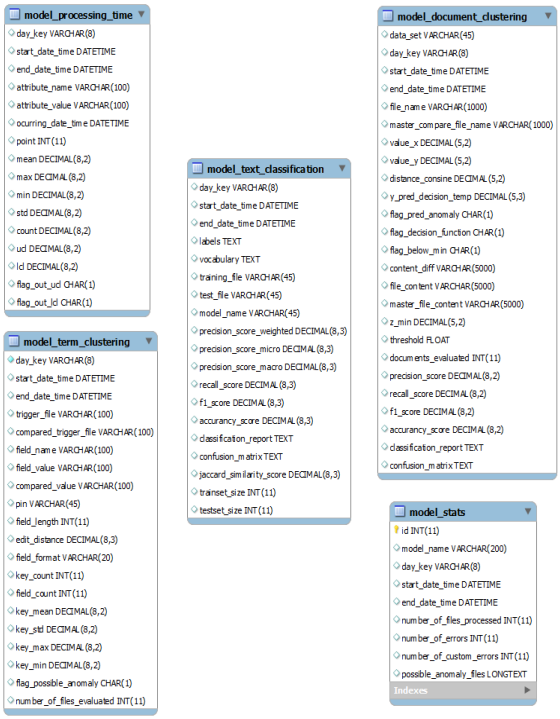

Figure 6
Database Tables Design

GUI Design

A simple GUI was designed. It permits the execution of models with basic knowledge of the Machine Learning techniques. See figure 7 for example of the designed GUI.

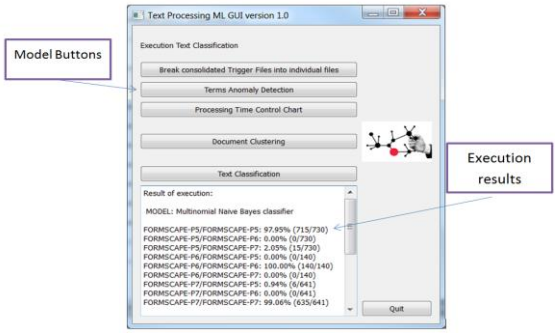

Figure 7
Graphic User Interface

PERFORMANCE OF THE MODELS

For supervised models designed for documents outlier detection and text classification the performance metrics are confusion matrix, recall score, precision score, f1 score, and accuracy score. The confusion matrix summarized the number of records correctly predicted or not. Recall and precision are used to measure if one class is considered significant than other classes. See the figure 8 is displayed the documents outlier detection chart. The Documents Outlier Detection model has better results if the training set doesn't include any outlier document because the training set is used to determinate the linear learner frontier. The linear learner frontier is utilized to determinate is a value is an inlier (value above the line) or outlier (value below the line). Any outlier document should be included in the test set. In the case using linear using OneClassSVM can identify the outlier documents with the flag_pred_anomaly equal to Y. A classification report and confusion report metrics are computed for this model. The anomaly values are the points below of the orange triangle. See Figure 8.

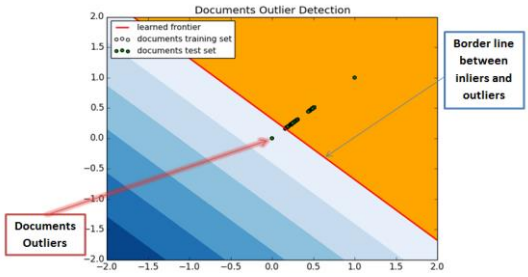

Figure 8
Documents Outlier Detection

The text classification execution includes the confusion matrix for the Naïve and LinearSVM model execution. See the Figure 9 and 10. The performance of the text classification is determinate by metric accuracy, recall, f1 score, Jaccard similarity score, precision, confusion matrix and classification reports. Based on the execution of this model was that Linear SVC had better results compared to Naïve Bayes classifier. See Figure 10.

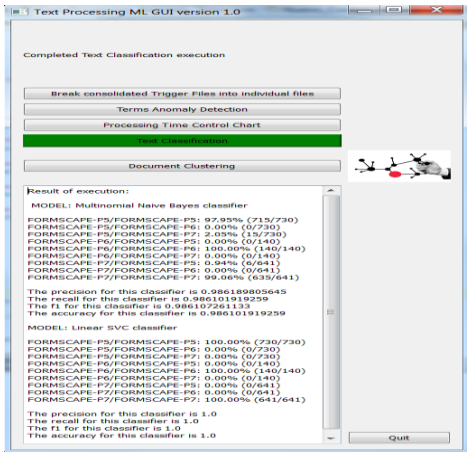

Figure 9
Text Classification Metrics Results

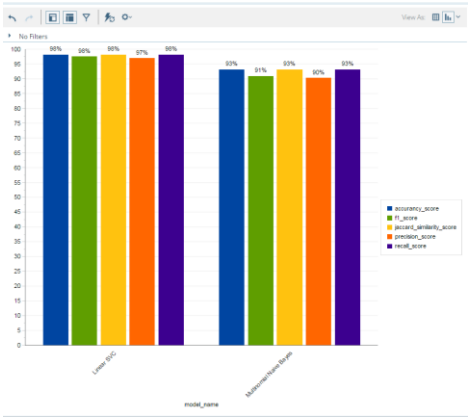

Figure 10
Text Classification Metrics Comparison Metrics Comparison Chart of LinearSVC and Naive Bayes Models

A term is considered anomalous when the weighted levenshtein distance result is greater or equal than 1 compared in pairwise for another value, both belonging same attribute name. See Figure 11. This model permits satisfactory detection of possible anomaly terms doing use of the weighted levenshtein distance algorithm. In the evaluation of the model was found the attribute values printer name and type of label have a normally characters and string size variation. Both attributes are considered not critical for the detection of anomalous values because any incorrect value will resulting that is not possible to print the label and the detectability of incorrect values in these attributes is high. Excluding these attributes will avoid few false positives. Nevertheless, in future work to enhance this model will be considered to include the anomaly detection of the attributes if necessary. During the evaluation of this model was observed that same value one in lower case and another in upper case was detected by the levenshtein distance algorithm as an outlier. By standarization the lower case values should be considered as anomalous. To measure the performance of this unsupervised model could the numbers of possible outlier terms of unlabeled data set divided by the total of analyzing terms. Giving the rate of anomalous values; result of more than 1% will be an indicator of some going wrong with the model or a lot of noise in the data. Requiring a further investigation of the issue. Using classification report and confusion matrix are available on the partially developed model to perform decision tree based on the entropy/information gain using Decision Tree classifier from sklearn package. The partially Decision Tree model design provides results, but it is not considered completed because of it requires to be validated and was not possible as part of this project by time constraint. Detailed information of the outlier terms are represented in a tabular format. See Figure 12.

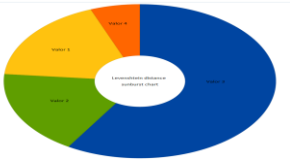

Figure 11
Terms Anomaly Detection Chart

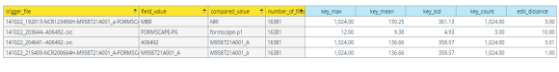

Figure 12
Outlier Terms Raw Data Table

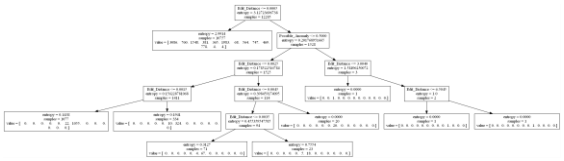

Figure 13
Decision Tree on Analyzed Data using Edit Distance Algorithm and Classification as Outlier or Not

The processing time control chart (SPC) provides a visualization of the execution using subplots for printers chart and another chart for label template categories. Using this control chart considering the data as a normal distribution in this unsupervised model is able to detect the variability in the processing of the trigger files. In the visualization the central line value and an upper control limit (UCL) are displayed, both dynamically calculated based on the data analyzed. See Figure 1. See Figure 15 for the UCL formula. For the Processing Time Control Chart model performance the mean and standard deviation from the unsupervised data analysis are used to detect inlier and outlier points. Values above the upper control limit (UCL) are marked as anomalous processing time.

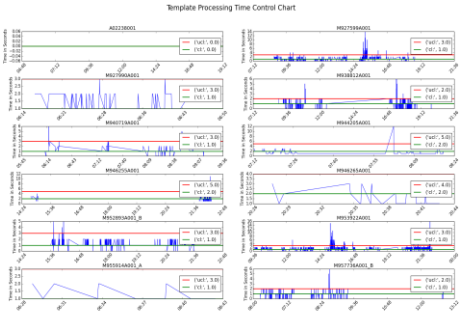

Figure 14
Processing Time by Template Control Chart (SPC)

```
g_ucl = round(float(grp[column_axes].mean() + (grp[column_axes].std() * 3)))
```

Figure 15
Compute the Upper Control Limit (UCL)

FUTURE WORK

Enhance the design of the models to include integration with Big Data analysis of the trigger files across multiple company business units. Research of another approach for term anomaly detection based on the distribution of the key and value combination. Evaluation of other algorithms such Artificial Neural Network and model parameterization available from the user interface execution. Evaluate the viability of real time anomaly detection. Enhance the term anomaly detection model and completion of decision tree to automatically perform feature selection to be more tolerant to noisy data.

CONCLUSION

This project provided a proof of concept that demonstrated the viability of anomaly detection on text document using Machine Learning techniques information retrieval, data mining and text mining. This system has been developed using of Python platform, sklearn , pandas and matplotlib packages [12]. It is demonstrated that it is possible to detect anomaly values assuming they are far other values. Doing use of the documents outlier model is able to detect anomalous document found in the test data set. Terms anomaly detection is using the text mining techniques TF-IDF, Levenshtein distance and NLTK, including output in which files the anomalous term values were found. Text classification model permits to predict “Template” class using Naïve Bayes and LinearSVC techniques. For supervised model text classification and document outlier models the metrics classification report and confusion report are used to measure the performance of the model. Using (SPC) control chart model is able detect anomalies in Processing Time execution of the trigger files. This project demonstrated proof of concept and is a major step towards a usable trigger file, text and processing time anomaly detection.

REFERENCES

[1] Guthrie, D.,“Unsupervised Detection of Anomalous Text”, University of Sheffield, [Online], July, 2008. Retrieved from: http://nlp.shef.ac.uk/Completed_PhD_Projects /guthrie.pdf.

[2] Chiwara, M., Al-Ayyoub, M., Hossain, M., S. and Gupta, R., “CSE 634 – Data Mining Text Mining”, Stony Brook

University, [Online], July, 2004. Retrieved from: http://www3.cs.stonybrook.edu/~cse634/presentations/TextMining.pdf.

[3] Leskovec, J., Rajaraman, A., and Ullman, J. D., "Mining of Massive Datasets", Stanford Univ., [Online], July 2014, Retrieved from: http://infolab.stanford.edu/~ullman/mmds/book.pdf.

[4] Huang, H., "Rank Based Anomaly Detection Algorithms", Syracuse University, [Online], March, 2013, Retrieved from: http://surface.syr.edu/cgi/viewcontent.cgi?article=1335&context=eecs_etd.

[5] Banerjee, A., Chandola, V., Kumar, V., Srivastava, J., and Lazarevic, A., "Anomaly Detection: A Tutorial", University of Minnesota, [Online], February, 2008, Retrieved from: https://www.siam.org/meetings/sdm08/TS2.ppt.

[6] Perkins, J., "Python Text Processing with NLTK 2.0 Cook book", January, 2010, Birmingham, UK: Pack Publishing. [Online], Retrieved from: https://www.packtpub.com/sites/default/files/3609-chapter-3-creating-custom-corpora.pdf.

[7] Kothari, P., "Cooking Python: Text Mining Curry NLTK 2.0", *UK: Pack Publishing*, 2010.

[8] Lama, P., "Clustering System Based On Text Mining Using the K-Means Algorithm", Tukku University of Applied Sciences, [Online], 2013. Retrieved from: http://www.theseus.fi/bitstream/handle/10024/69505/Lama_Prabin.pdf?sequence=1.

[9] Byrd, S., Klein, E., and Loper, E., "Natural Language Processing with Python", Sebastopol, California: O'Reilly, 2009.

[10] Domingos, P., "A Few Useful Things to Know about Machine Learning", University of Washington, [Online], 2012, Retrieved from: http://homes.cs.washington.edu/~pedrod/papers/cacm12.pdf.

[11] Nining, P., Steinbach, M. and Kumar, V., "Introduction to Data Mining", New Jersey: Pearson Education, Inc, 2006, pp. 2 – 12.

[12] Garreta, R., and Moncecchi, G., "Learning scikit-learn: Machine Learning in Python", Birmingham, UK: Pack Publishing, 2013.

[13] Marsteller, J., Marsteller, Dr., "Advanced Methods in Delivery System Research – Planning, Executing, Analyzing, and Reporting Research on Delivery System Improvement Webinar #2: Statistical Process Control", *Agency for Healthcare Research and Quality*, [Online], March 3, 2012. Retrieved from: http://www.ahrq.gov/professionals/prevention-chronic-care/improve/coordination/webinar03/logic_models.pptx.